DATABASE MANAGEMENT SYSTEM : NEWRUN

PROJECT REPORT

Varun Mohankumar Jayasree - Z1983582

Shashank Sangu - Z1974994

Rohith Irvichetty - Z1995294

Deepak Kolathur – Z1980621

OPERATIONS AND MANAGEMENT INFORMATION SYSTEMS

NORTHERN ILLINOIS UNIVERSITY

OMIS-652 DATABASE MANAGEMENT SYSTEMS

PROFESSOR - KISHEN IYENGAR

May 1st, 2024

Project Description:

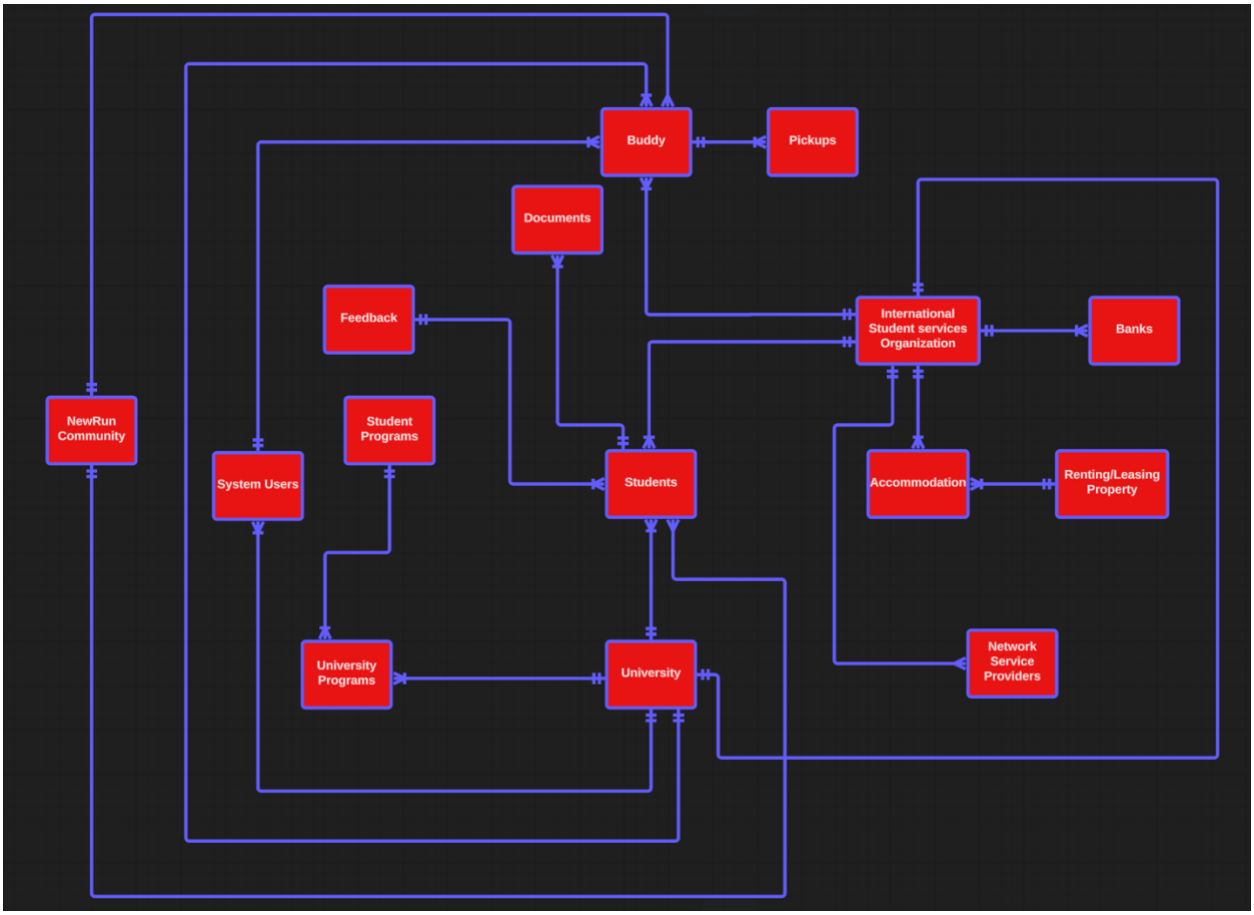
NewRun is a dedicated firm exclusively catering to the needs of newly arriving international students in the USA. Recognizing the challenges faced by these students upon their arrival, NewRun offers a comprehensive range of essential services to facilitate a smooth transition. From Airport pickup to securing accommodation and setting up vital services like banking and network connectivity, NewRun ensures that international students receive the support they need to settle into their new environment seamlessly.

The process begins with international students providing their data online to NewRun. Upon collecting the information, NewRun's software coordinates with the relevant bodies of universities across the USA to ensure a tailored and efficient service. This involves arranging for volunteers, known as buddies, to pick up students from the airport and transport them to their respective universities/ accommodations. Simultaneously, NewRun assists students in securing suitable accommodation, coordinating with local housing properties based on individual preferences and needs. Furthermore, NewRun facilitates the process of setting up essential services such as opening a bank account, obtaining a SIM card, and applying for a Social Security Number (SSN), ensuring that students have access to the necessary resources to navigate life in the USA smoothly.

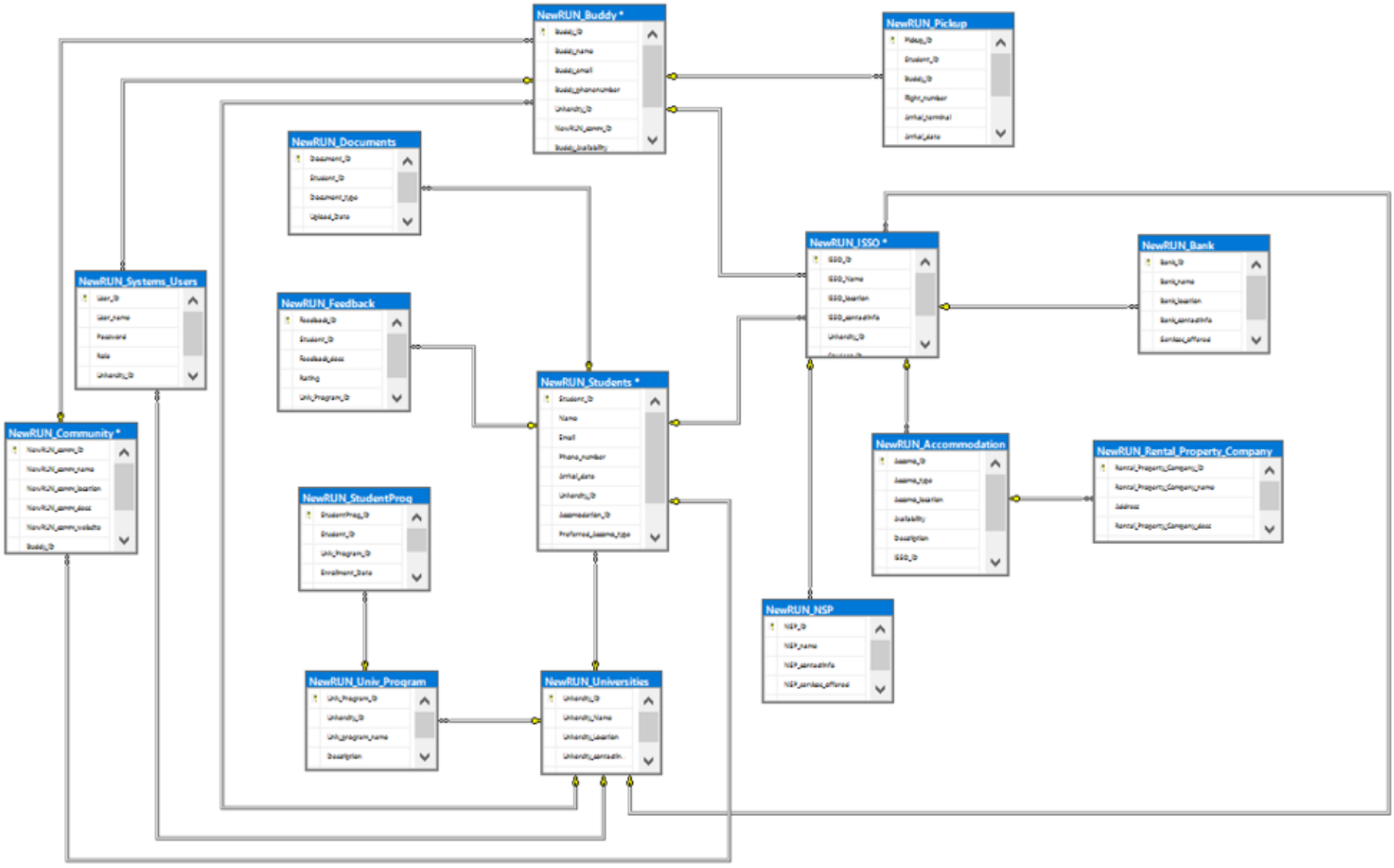
To effectively manage and coordinate these services, NewRun's database can be structured around key entities relevant to its operations. The primary entity is Students, for which a table can be created to store student information including IDs, contact details, arrival dates, and specific needs such as accommodation preferences and banking requirements. Another crucial entity is Universities, with a table dedicated to storing university details such as IDs, names, locations, and contact information. Accommodation represents another vital aspect, and a table can be created to manage accommodation details such as IDs, types, locations, and availability. Additionally, tables for Banks and Network Service Providers can store relevant information such as IDs, names, services offered, and contact details, facilitating assistance in setting up banking and network services. Finally, a table for Volunteers/Buddies can store information such as IDs, names, contact details, and availability, enabling efficient coordination between volunteers and students for transportation assistance and other needs upon arrival. Through this structured approach, NewRun ensures that international students receive comprehensive support tailored to their specific requirements as they embark on their academic journey in the USA.

List of Tables:

E-R Diagram for NewRUN Project



Database Diagram for NewRUN Project



Queries

1. Listing the number of students who are currently using Dorm facilities in the enlisted Universities:

Solution:

```
SELECT u.university_Name,s.Preferred_Accomo_type, count(s.student_ID) AS Number_of_students

FROM NewRUN_Students s

INNER JOIN NewRUN_Universities u ON u.University_ID = s.University_ID

WHERE s.Preferred_Accomo_type like '%dorm%'

GROUP BY u.University_Name,s.Preferred_Accomo_type
```

Output:

Results | Messages

	university_Name	Preferred_Accomo_type	Number_of_students
1	Northern Illinois University	Dorm (Double Occupancy)	1
2	Illinois State University	Dorm (Single Occupancy)	1
3	Northwestern University	Dorm (Suite)	1
4	DePaul University	Dorm Apartment	1

2. Which Student have placed the highest number of feedbacks:

Solution:

```
SELECT TOP 1 f.Student_ID,s.name,s.Email,COUNT(f.student_ID)as Count_of_Feedbacks

FROM NewRUN_Feedback f

INNER JOIN NewRUN_Students s ON s.Student_ID =f.Student_ID

GROUP BY f.Student_ID,s.name,s.Email

ORDER BY Count_of_Feedbacks DESC
```

Results | Messages

	Student_ID	name	Email	Count_of_Feedbacks
1	1002	David Walker	david.walker@email.com	3

3. Retrieve all students who have completed their programs along with their feedback:

Solution:

```
SELECT s.Student_ID, s.Name, s.Email, s.Phone_number, s.Arrival_date,
s.University_ID,F.Feedback_ID, F.Rating,F.Date_posted

FROM NewRUN_Students s

JOIN NewRUN_Feedback F ON s. Student_ID = F. Student_ID

JOIN NewRUN_StudentProg SP on s.Student_ID= SP.Student_ID

WHERE SP.Completion_Status= 'Completed'
```

Output:

Results Messages

	Student_ID	Name	Email	Phone_number	Arrival_date	University_ID	Feedback_ID	Rating	Date_posted
1	1004	Michael Young	michael.young@email.com	+86 188 1234 5678	8/20/24	2001	9004	5	2024-03-11 00:00:00.000
2	1006	William Lee	william.lee@email.com	+81 3 1234 5678	8/20/24	2002	9006	5	2020-02-23 00:00:00.000

4. Number of buddies present in each Universities associated with NewRUN:

Solution:

```
SELECT U.University_Name, COUNT(B.Buddy_ID) AS Buddy_Count

FROM NewRUN_Universities U

LEFT JOIN NewRUN_Buddy B ON U.University_ID = B.University_ID

GROUP BY U.University_Name
```

Output:

Results Messages

	University_Name	Number_of_Buddies
1	DePaul University	1
2	Illinois State University	0
3	Loyola University Chicago	4
4	Northern Illinois University	3
5	Northwestern University	1
6	Southern Illinois University Carbondale	1
7	University of Illinois Urbana-Champaign	1

5. Find the buddy details for students arriving on a particular date along with their pickup information:

Solution:

SELECT b.Buddy_ID,b.Buddy_Name, b.Buddy_Email, b.Buddy_PhoneNumber,
b.University_ID,p.Flight_number, p.Arrival_terminal,p. Arrival_date,
p.Arrival_time,p.Student_Contac

FROM NewRUN_Buddy b

JOIN NewRUN_Pickup p ON b. Buddy_ID = p. Buddy_ID

WHERE p. Arrival_date = '2/1/23'

Output:

Results | Messages

	Buddy_ID	Buddy_Name	Buddy_Email	Buddy_PhoneNumber	University_ID	Flight_number	Arrival_terminal	Arrival_date	Arrival_time	Student_Contact_number
1	7001	jagadeesh_dollipalla	jd@niu.edu	5692748769	2001	AA1234	Terminal 3	2/1/23	11:14 AM	8157648862

6. Find the details of all students who have uploaded their passports along with their program information:

Solution:

Select s.Student_ID, s. Name, s. Email, s. Phone number, s. Arrival_date, s.University_ID, s.Passport number, d. Document_Type, d.Upload_Date

From NewRUN_Students s

JOIN NewRUN_Documents d ON s.Student_ID = d.Student_ID

WHERE d. Document_Type = 'Passport' AND s. Student_ID IN (

SELECT Student_ID

FROM NewRun_StudentProg

WHERE Completion_ Status = 'Completed')

Output:

	Student_ID	Name	Email	Phone_number	Arrival_date	University_ID	Passport_number	Document_Type	Upload_Date
1	1006	William Lee	william.lee@email.com	+81 3 1234 5678	8/20/24	2002	JP1234567	Passport	2024-07-26 00:00:00.000

7. Retrieve the names of all students along with the universities they are enrolled in and the corresponding ISSO contact information:

Solution:

```
SELECT s.Name AS Student_Name, u.University_Name, u.University_location,
    iso.ISSO_Contactinfo

FROM NewRUN_Students s

JOIN NewRUN_Universities u ON s.University_ID = u.University_ID

JOIN NewRUN_ISSO iso ON s.University_ID = iso.University_ID
```

Output:

	Student_Name	University_Name	University_location	ISSO_Contactinfo
1	Alice Johnson	Northern Illinois University	DeKalb, Illinois	ISSS@uiuc.edu
2	Alice Johnson	Northern Illinois University	DeKalb, Illinois	ISSS@uiuc.edu
3	David Walker	Illinois State University	Normal, Illinois	ISSS@niu.edu
4	David Walker	Illinois State University	Normal, Illinois	ISSS@niu.edu
5	Emily Garcia	DePaul University	Chicago, Illinois	ISSS@luc.edu
6	Michael Young	Northern Illinois University	DeKalb, Illinois	ISSS@uiuc.edu
7	Michael Young	Northern Illinois University	DeKalb, Illinois	ISSS@uiuc.edu
8	Sophia Hernandez	Loyola University Chicago	Chicago, Illinois	ISSS@siuc.edu
9	William Lee	University of Illinois Urbana-Champaign	Urbana, Illinois	ISSS@nwu.edu

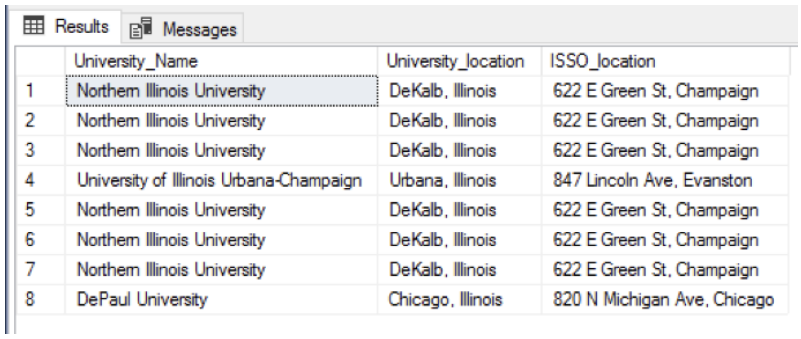
8. Find all universities and their ISSO locations where the buddy speaks a language other than English.

Solution:

```
SELECT u.University_Name, u.University_location, iso.ISSO_location

FROM NewRUN_Universities u

JOIN NewRUN_ISSO iso ON u.University_ID = iso.University_ID

JOIN NewRUN_Buddy b ON u.University_ID = b.University_ID
```

```
WHERE
        b.Buddy_ID IN (
        SELECT Buddy_ID
        FROM NewRUN_Buddy
        WHERE Language <> 'English')
```

Output:



Results | Messages

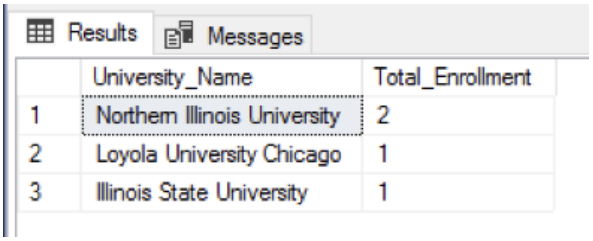
	University_Name	University_location	ISSO_location
1	Northern Illinois University	DeKalb, Illinois	622 E Green St, Champaign
2	Northern Illinois University	DeKalb, Illinois	622 E Green St, Champaign
3	Northern Illinois University	DeKalb, Illinois	622 E Green St, Champaign
4	University of Illinois Urbana-Champaign	Urbana, Illinois	847 Lincoln Ave, Evanston
5	Northern Illinois University	DeKalb, Illinois	622 E Green St, Champaign
6	Northern Illinois University	DeKalb, Illinois	622 E Green St, Champaign
7	Northern Illinois University	DeKalb, Illinois	622 E Green St, Champaign
8	DePaul University	Chicago, Illinois	820 N Michigan Ave, Chicago

9. Retrieve the top 3 universities with the highest enrollment for undergraduate programs

Solution:

```
SELECT TOP 3 u.University_Name, COUNT(s.Student_ID) AS Total_Enrollment
FROM NewRUN_Universities u
JOIN NewRUN_Students s ON u.University_ID = s.University_ID
JOIN NewRun_StudentProg sp ON s.Student_ID = SP.Student_ID
WHERE SP.Univ_Program_ID LIKE '400%'
GROUP BY u.University_Name
ORDER BY Total_Enrollment DESC
```

Output:



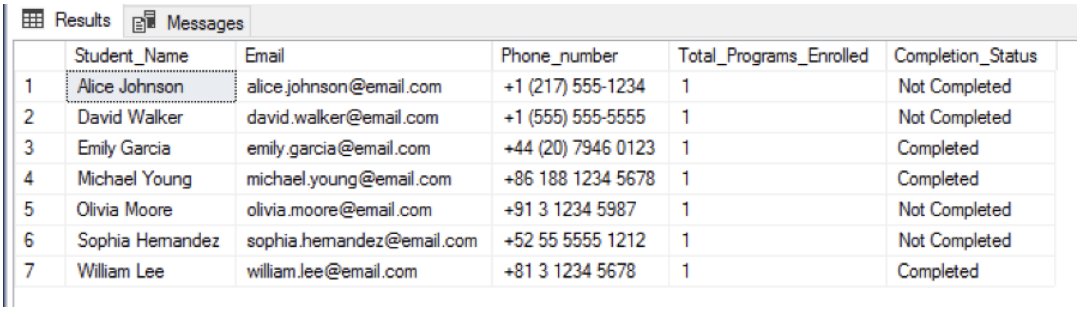
	University_Name	Total_Enrollment
1	Northern Illinois University	2
2	Loyola University Chicago	1
3	Illinois State University	1

10. Find all students along with the number of programs they are enrolled in and their completion status.

Solution:

```
SELECT s.Name AS Student_Name, s.Email, s.Phone_number, COUNT (DISTINCT SP.Univ_Program_ID) AS Total_Programs_Enrolled,
        CASE
        WHEN MAX(SP.Completion_Status) = 'Completed' THEN 'Completed'
        ELSE 'Not Completed'
         END AS Completion_Status
FROM NewRUN_Students s
LEFT JOIN NewRun_StudentProg SP ON s.Student_ID = SP.Student_ID
GROUP BY s.Name, s.Email, s.Phone_number;
```

Output:



	Student_Name	Email	Phone_number	Total_Programs_Enrolled	Completion_Status
1	Alice Johnson	alice.johnson@email.com	+1 (217) 555-1234	1	Not Completed
2	David Walker	david.walker@email.com	+1 (555) 555-5555	1	Not Completed
3	Emily Garcia	emily.garcia@email.com	+44 (20) 7946 0123	1	Completed
4	Michael Young	michael.young@email.com	+86 188 1234 5678	1	Completed
5	Olivia Moore	olivia.moore@email.com	+91 3 1234 5987	1	Not Completed
6	Sophia Hernandez	sophia.hernandez@email.com	+52 55 5555 1212	1	Not Completed
7	William Lee	william.lee@email.com	+81 3 1234 5678	1	Completed

11. Popularity of the University Program based on the number of students enrolled ranking lowest to highest.

Solution:

```
SELECT UP.Univ_program_name,U.University_Name,COUNT(SP.Student_ID) AS Student_Count

FROM NewRUN_Univ_Program UP

JOIN NewRUN_Universities U ON UP.University_ID = U.University_ID

LEFT JOIN NewRun_StudentProg SP ON UP.Univ_Program_ID = SP.Univ_Program_ID

GROUP BY U.University_Name, UP.Univ_program_name

ORDER BY Student_Count
```

Output:

Results Messages

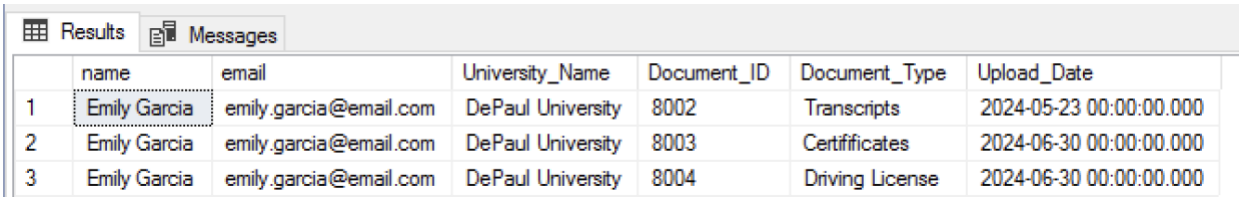
	Univ_program_name	University_Name	Student_Count
1	Accounting (B.S.)	Northern Illinois University	0
2	Mechanical Engineering (B.S.)	University of Illinois Urbana-Champaign	0
3	Psychology (B.S.)	Loyola University Chicago	1
4	Data Science (M.S.)	DePaul University	1
5	Electrical Engineering (B.S.)	University of Illinois Urbana-Champaign	1
6	Business Administration (M.B.A.)	University of Illinois Urbana-Champaign	2
7	Computer Science (B.S.)	Northern Illinois University	2

12. Listing the available of documents for a specific student from a specific University(DePaul University):

Solution:

```
SELECT s.name,s.email,u.University_Name,D.Document_ID,D.Document_Type,D.Upload_Date

FROM NewRUN_Documents D

JOIN NewRUN_Students S ON D.Student_ID = S.Student_ID

JOIN NewRUN_Universities U on s.University_ID = u.University_ID

WHERE S.Student_ID = 1003 AND u.University_Name LIKE '%Depaul%'
```

Output:



Results | Messages

	name	email	University_Name	Document_ID	Document_Type	Upload_Date
1	Emily Garcia	emily.garcia@email.com	DePaul University	8002	Transcripts	2024-05-23 00:00:00.000
2	Emily Garcia	emily.garcia@email.com	DePaul University	8003	Certifificates	2024-06-30 00:00:00.000
3	Emily Garcia	emily.garcia@email.com	DePaul University	8004	Driving License	2024-06-30 00:00:00.000

13. Getting Buddies info based on preferred language or from a certain Community belonging to a specific University:

Solution:

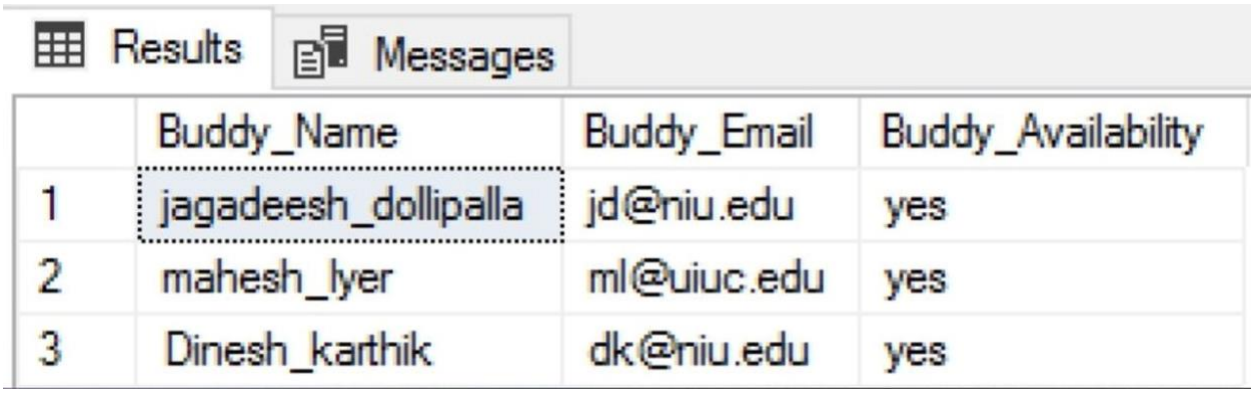
SELECT b.Buddy Name, b.Buddy_Email, b-Buddy_Availability

FROM NewRUN Buddy b

INNER JOIN NeWRUN_Universities u ON b.University_ID = u.University_ID

WHERE .University_ID = 2001 OR (b.Language LIKE %Tamil%' OR b.NewRUN_comm_ID = 13004)

Output:



Results | Messages

	Buddy_Name	Buddy_Email	Buddy_Availability
1	jagadeesh_dollipalla	jd@niu.edu	yes
2	mahesh_Iyer	ml@uiuc.edu	yes
3	Dinesh_karthik	dk@niu.edu	yes

14. Popular Accommodation Types by University:

Solution:

SELECT u.University Name AS University, a.Accomo_type, COUNTs.Student ID FROM NewRUN Universities u

INNER JOIN NeWRUN_Students s ON u.University_TD = s.University_TD

INNER JOIN NEWRUN Accommodation a s.Accomo_ID = a. Accomo, TD

GROUP BY u.University Name, a. Accomo_type

ORDER BY COUNT (s. Student ID) DESC

Output:

Results Messages

	University	Accomo_type	Count_of_choices
1	Northern Illinois University	TownHouse	2
2	Northwestern University	TownHouse	1
3	University of Illinois Urbana-Champaign	TownHouse	1
4	DePaul University	Appartment	1
5	Illinois State University	Appartment	1
6	Loyola University Chicago	Appartment	1

15. Listing the count of active students in various NewRUN Communities:

Solution:

SELECT c.NewRUN_comm_name, COUNT (s.Student_ TD) AS Active Students

FROM NewRUN_Community c

LEFT JOIN NEWRUN_Students s ONc. Student_ ID = s.Student TD

GROUP BY c.NewRUN_comm_name

ORDER BY Active_Students DESC

Output:

Results Messages

	NewRUN_comm_name	Active_Students
1	IPL Fan club	3
2	UIUC Foodie Group	1
3	UIUC Hiking & Outdoors Club	1
4	UIUC International Movie Nights	1
5	Illinios International Student Forum	1

16. Available Banks and their Services near the Universities

Solution:

SELECT u. University Name, b. Bank_ name, b. Services offered

FROM NewRUN_ Bank b

INNER JOIN NEWRUN_ISSO isso ON b. ISSO_TD = isso. ISSO ID

INNER JOIN NewRUN_ Students s ON isso. STUDENT_ID= S. Student TD

INNER JOINNewRUN_Universities u ON s.University_ID = u. University_ ID

GROUP BY b. Bank_name, b.Services_offered, u.University_name, u.University_Location

ORDER BY u. University_name

Output:

Results Messages

	University_Name	Bank_name	Services_offered
1	DePaul University	The Community Credit Union	Online Banking
2	Illinois State University	City Bank	Savings
3	Loyola University Chicago	Chase Bank	Savings
4	Northern Illinois University	Bank of America	Investment Accounts
5	Northern Illinois University	First National Bank	Checking
6	Northwestern University	US Bank	Investment Accounts
7	University of Illinois Urbana-Champaign	Wells Fargo	Checking

17.Find the programs with enrollment dates after August 1, 2024, along with the names of students enrolled in those programs.

Solution:

SELECT s.Name AS Student_Name,up.Univ_program_name AS Program_Name, sp.Enrollment_Date

FROM NewRUN_Students s

JOIN NewRun_StudentProg sp ON s.Student_ID = sp.Student_ID

JOIN NewRUN_Univ_Program up ON sp.Univ_Program_ID = up.Univ_Program_ID

WHERE sp.Enrollment_Date > '2024-08-01'

Output:

Results Messages

	Student_Name	Program_Name	Enrollment_Date
1	Alice Johnson	Computer Science (B.S.)	8/2/2024
2	David Walker	Computer Science (B.S.)	8/2/2024
3	Emily Garcia	Business Administration (M.B.A.)	8/10/2024
4	Michael Young	Electrical Engineering (B.S.)	8/13/2024
5	Sophia Hernandez	Business Administration (M.B.A.)	8/2/2024
6	William Lee	Psychology (B.S.)	8/10/2024
7	Olivia Moore	Data Science (M.S.)	8/15/2024

18. Find the universities along with the total number of students enrolled in programs with a maximum enrollment greater than 200.

Solution:

```
SELECT u.University_Name, COUNT(s.Student_ID) AS Total_Students
FROM NewRUN_Universities u
JOIN NewRUN_Students s ON u.University_ID = s.University_ID
JOIN NewRUN_Univ_Program UP ON s.University_ID = UP.University_ID
WHERE UP.Maximum_Enrollment > 200
GROUP BY u.University_Name;
```

Output:

Results Messages

	University_Name	Total_Students
1	Loyola University Chicago	1
2	Northern Illinois University	2
3	University of Illinois Urbana-Champaign	3

19. Retrieve all students who have completed their programs and have provided feedback with a rating greater than the average rating.

Solution:

```
SELECT s.Student_ID,s.Name,s.Email,f.Feedback_desc,f.Rating,f.Date_posted

FROM NewRUN_Students s

JOIN NewRUN_Feedback f ON s.Student_ID = f.Student_ID

WHERE s.Student_ID IN (

    SELECT Student_ID

    FROM NewRun_StudentProg

    WHERE Completion_Status = 'Completed')

  AND f.Rating > (

    SELECT AVG(Rating)

    FROM NewRUN_Feedback)
```

Output:

Results | Messages

	Student_ID	Name	Email	Feedback_desc	Rating	Date_posted
1	1004	Michael Young	michael.young@email.com	I would recommend this course to others.	5	2024-03-11 00:00:00.000
2	1006	William Lee	william.lee@email.com	The class discussions were very helpful.	5	2020-02-23 00:00:00.000

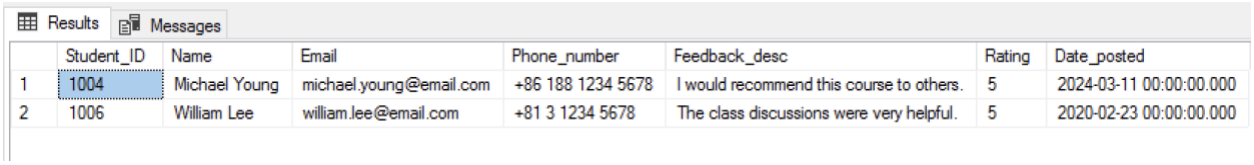
20. Retrieve all students who have completed their programs and have provided feedback with a rating greater than 3.

Solution:

```
SELECT s.Student_ID,s.Name, s.Email, s.Phone_number, f.Feedback_desc,f.Rating,
f.Date_posted

FROM NewRUN_Students s

JOIN NewRUN_Feedback f ON s.Student_ID = f.Student_ID

WHERE s.Student_ID IN (

    SELECT Student_ID

    FROM NewRun_StudentProg
```

```
WHERE Completion_Status = 'Completed') AND f.Rating > 3
```

Output:



Results | Messages

	Student_ID	Name	Email	Phone_number	Feedback_desc	Rating	Date_posted
1	1004	Michael Young	michael.young@email.com	+86 188 1234 5678	I would recommend this course to others.	5	2024-03-11 00:00:00.000
2	1006	William Lee	william.lee@email.com	+81 3 1234 5678	The class discussions were very helpful.	5	2020-02-23 00:00:00.000